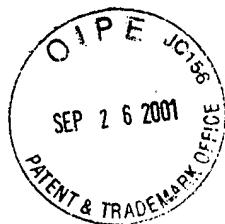




Europäisches  
Patentamt

European  
Patent Office

Office européen  
des brevets



Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

00116484.7

Der Präsident des Europäischen Patentamts:  
Im Auftrag

For the President of the European Patent Office  
Le Président de l'Office européen des brevets  
p.o.

MÜNCHEN, DEN  
MUNICH, 14/08/01  
MUNICH, LE



Europäisches  
Patentamt

European  
Patent Office

Office européen  
des brevets

**Blatt 2 der Bescheinigung**  
**Sheet 2 of the certificate**  
**Page 2 de l'attestation**

Anmeldung Nr.:  
Application no.: **00116484.7**  
Demande n°:

Anmeldetag:  
Date of filing: **29/07/00**  
Date de dépôt:

Anmelder:  
Applicant(s):  
Demandeur(s):  
**main Gesellschaft für Informationsverarbeitung mbH**  
**70174 Stuttgart**  
**GERMANY**

Bezeichnung der Erfindung:  
Title of the invention:  
Titre de l'invention:  
**Kommunikationsverfahren zwischen Server und Client eines Netzwerkes unter Zuhilfenahme von  
Zustandsinformationen**

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: Tag: Aktenzeichen:  
State: Date: File no.  
Pays: Date: Numéro de dépôt:

Internationale Patentklassifikation:  
International Patent classification:  
Classification internationale des brevets:  
**G06F17/30, H04L29/06**

Am Anmeldetag benannte Vertragstaaten:  
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR  
Etats contractants désignés lors du dépôt:

Bemerkungen:  
Remarks:  
Remarques:

**Ursprünglicher Titel: Siehe Seite 1 der Beschreibung**

EPO - Munich  
20  
29. Juli 2000

Stuttgart, 28.07.2000  
P7552EP Rk/pa

Anmelder:

main Gesellschaft für Informationsverarbeitung mbH  
Lange Straße 54  
D-70174 Stuttgart

Vertreter:

Kohler Schmid + Partner  
Patentanwälte GbR  
Ruppmannstraße 27  
D-70565 Stuttgart

Kommunikationsverfahren zwischen Server und Client  
eines Netzwerks

Die Erfindung betrifft ein Kommunikationsverfahren zwischen einem Netzwerk-Client und einem Netzwerk-Server, wobei ein vom Netzwerk-Client angefragtes Netzwerk-Dokument vom Netzwerk-Server an den Netzwerk-Client geschickt und dann von einem Browser des Netzwerk-Clients dem Benutzer zur Bearbeitung angezeigt wird, sowie ein zum Durchführen dieses Kommunikationsverfahrens geeignetes Computerprodukt.

Die Quelle von Dokumenten im Internet ist der sogenannte Server. Im World Wide Web (WWW) werden die WWW- bzw. Web-Seiten von einem Web-Server geliefert, wobei das Auffinden von Web-Seiten im Internet mit Hilfe von Hinweis-Zeigern erfolgt, die mit HTML (engl. Hypertext Markup Language) in die gespeicherten Dokumente eingebracht werden. Die gelieferte Web-Seite wird dem Benutzer mittels eines Web-Browsers (d.i. das Werkzeug zum Navigieren und Modifizieren im Internet) angezeigt.

Eine Anfrage eines Internet-Benutzers im Internet erfolgt prinzipiell folgendermaßen:

1. Der Benutzer richtet über seinen Netzwerk-Rechner (sog. Client) eine HTTP-Anfrage (HTTP: HyperText Transfer Protokoll) an den Web-Server.
2. Der Web-Server lädt die angefragte Web-Seite über ein Filesystem oder erzeugt die Web-Seite auf andere Art und Weise.
3. Diese Web-Seite wird an den Client via HTTP zurückgeschickt.
4. Der Web-Browser des Client zeigt die Web-Seite an...
5. Der Benutzer arbeitet mit der Web-Seite.

Allerdings gehen Daten, die der Benutzer eingibt, ohne weitere technische Maßnahmen verloren. HTTP ist ein zustandsloses Protokoll. Das beschriebene Verfahren ist nur begrenzt "individualisierbar", da alle Clients im Prinzip die selbe Web-Seite verwenden. Beim erneuten Aufruf einer Web-Seite

wird dem Benutzer daher die Web-Seite im stets gleichen Layout und ohne Berücksichtigung von Eingaben oder Änderungen, die vom Benutzer bei seinem letzten Aufruf bereits gemacht worden sind, angezeigt. D.h., die Generierung der angefragten Web-Seite erfolgt statisch.

Damit die Daten, die der Benutzer eingibt, nicht verloren gehen, sind bereits Techniken wie z.B. Cookies oder Abspeichern von Formulardaten in einer Datenbank bekannt. Diese Speicherung bezieht sich in jedem Fall nur auf die "nackten" Daten (z.B. die Eingaben in Felder), nicht jedoch auf den Zustand der gesamten Seite, der neben den nackten Daten auch die Vorgängerseite, Schriftart/-größe, Position der Elemente und des Cursors etc. beinhaltet.

Es ist daher die Aufgabe der Erfindung, ein Verfahren der eingangs genannten Art derart zu verbessern, daß beim erneuteten Aufruf eines Netzwerk-Dokuments durch den gleichen Benutzer alle vom Benutzer bei einem früheren Aufruf am Netzwerk-Dokument vorgenommenen Änderungen und Eingaben bereits berücksichtigt werden, sowie ein entsprechendes Computerprodukt bereitzustellen. Dabei sind unter Änderungen und Eingaben neben eingegebenen Daten auch Änderungen am Layout (textueller Inhalt, Schriftarten, Positionen der Elemente, Farben, Hintergrund; Bilder etc.) sowie die Einbettung der Seite in den Gesamtkontext (z.B. Vorgängerseiten und mögliche Nachfolgerseiten) zu verstehen.

Diese Aufgabe wird erfindungsgemäß dadurch gelöst, daß die vom Benutzer an dem Netzwerk-Dokument am Netzwerk-Client durchgeföhrten Änderungen und Ergänzungen am Netzwerk-Client über Software aufgezeichnet und im Netzwerk-Server als benutzerspezifische Daten gespeichert werden und daß beim er-

neuten Aufruf des Netzwerk-Dokuments durch den Benutzer das bei einem früheren Aufruf am Netzwerk-Client durch die Interaktion des Benutzers erstellte Netzwerk-Dokument anhand den gespeicherten benutzerspezifischen Daten wiederhergestellt wird. Bezuglich des Internets bedeutet dies, daß sich der gesamte Zustand einer Web-Seite, die ein Benutzer durch seine Interaktion mit dem Netzwerk-Client zu einem früheren Zeitpunkt erzeugt hat, wiederherstellen lässt. Die wiederherzustellende Web-Seite wird dabei dynamisch erzeugt.

Mit diesem erfindungsgemäßen Verfahren ist es möglich, jedem Client beim erneuten Aufruf eines Netzwerk-Dokuments seine individuelle Netzwerk-Seite anzubieten. In der Regel handelt es sich dabei um die bei der letzten Sitzung des Benutzers am Netzwerk-Client generierten Seiten. Im Internet wird dadurch eine bisher nicht erreichte Individualisierung der Web-Seiten möglich. Erfindungsgemäß ist alles, was sich programmgesteuert im Web-Browser anlegen und/oder modifizieren lässt, wiederherstellbar. Das Internet bzw. die einzelnen Web-Seiten werden mandantenfähig, da jeder Client seine individuellen Daten und Einstellungen vorheriger Sitzungen wieder erhält. Alle bei der vorherigen Sitzung abgelaufenen Dialoge können sowohl bei der Wiederherstellung der Web-Seite als auch beim weiteren Dialog bzw. bei der weiteren Dialogführung mit dem Benutzer, z.B. hinsichtlich des Layouts von Dialogseiten, berücksichtigt werden.

Vorzugsweise werden die vom Benutzer am Netzwerk-Dokument am Netzwerk-Client durchgeführten Änderungen und Ergänzungen zunächst am Netzwerk-Client, insbesondere im Hauptspeicher oder auf der Festplatte, zwischengespeichert und erst, wenn die Bearbeitung des Netzwerk-Dokuments durch den Benutzer abgeschlossen ist, an den Netzwerk-Server übergeben. Für die

benutzerspezifischen Daten wird vom Netzwerk-Server in einer Datenbank vorteilhafterweise ein eigener Bereich anlegt.

Beim erstmaligen Aufruf eines Netzwerk-Dokuments durch einen Benutzer sollte die Identifikation des Netzwerk-Clients bzw. des Benutzers und z.B. die Vergabe eines Passworts erfolgen. Der Netzwerk-Server erzeugt für diesen Client einen eigenen Bereich in der Datenbank, in der die benutzerspezifischen Daten abgelegt werden. Dieser Bereich wird als "Push-Back-Bereich" bezeichnet. Sobald der Benutzer beim erneuten Kontakt identifiziert ist, können client-generierte Web-Seiten vom vorherigen Kontakt gezielt angezeigt werden.

Vorzugsweise wird das vom Benutzer geänderte bzw. ergänzte Dokument am Netzwerk-Client aufgezeichnet und im Netzwerk-Server in einem browserunabhängigen Format, insbesondere in einem browserunabhängigen XML-Format, JavaScript oder serialisierten Java-Objekten an den Netzwerk-Server übertragen und dort gespeichert. Alternativ können die aufgezeichneten Änderungen als reines HTML gespeichert werden, allerdings hat dies zur Folge, daß in der Regel kein Datenaustausch zwischen verschiedenen Browser-Versionen möglich ist.

Am Netzwerk-Client können die gespeicherten benutzerspezifischen Daten des Netzwerk-Servers über z.B. JavaScript oder Java interpretiert und daraus ein für den jeweiligen Browser lesbares Format (z.B. HTML-Code) dynamisch erzeugt werden.

Mit dem erfindungsgemäßen Verfahren am Netzwerk-Client können HTML-Seiten über Scriptsprachen und/oder Java generiert werden, die dann in einem Browser-neutralen Format an den Web-Server zurückgesendet werden. Die Generierung von kompletten HTML-Seiten am Netzwerk-Client und das Zurücksenden

dieser Seiten erlauben die Wiederherstellung des gesamten früheren Zustands einer Web-Seite. Dies stellt die Umkehrung des bisherigen Prinzips im Internet dar, bei dem HTML-Seiten am Netzwerk-Server gespeichert und/oder generiert und dann an den Netzwerk-Client gesendet werden. Bei dem erfindungsgemäßen Verfahren generiert der Netzwerk-Client neue HTML-Seiten, die dann an den Netzwerk-Server gesendet werden.

Die Erfindung betrifft auch ein Computerprodukt, das geeignet ist, alle Schritte des oben beschriebenen Kommunikationsverfahren durchzuführen. Dieses Computerprodukt kann ein Computerprogramm oder ein computerlesbares Speichermedium sein.

Weitere Vorteile der Erfindung ergeben sich aus der Beschreibung und der Zeichnung. Ebenso können die vorstehend genannten und die noch weiter aufgeführten Merkmale erfindungsgemäß jeweils einzeln für sich oder zu mehreren in beliebigen Kombinationen Verwendung finden. Die gezeigte und beschriebene Ausführungsform ist nicht als abschließende Aufzählung zu verstehen, sondern hat vielmehr beispielhaften Charakter für die Schilderung der Erfindung.

Es zeigt:

Fig. 1 schematisch den zeitlichen Ablauf des erfindungsgemäßen Kommunikationsverfahrens beim Erstkontakt zwischen einem Client und einem Web-Server; und

Fig. 2 schematisch den zeitlichen Ablauf des erfindungsgemäßen Kommunikationsverfahrens beim Zweitkontakt bzw. bei jedem weiteren Kontakt zwischen dem Client und dem Web-Server.

Fig. 1 beschreibt das Szenario, welches im Internet zwischen einem Client 1 und einem Web-Server 2 beim erstmaligen Kontakt zeitlich abläuft.

Der Client 1 stellt eine HTTP-Anfrage nach einer Web-Seite an einen Web-Server 2 (Schritt S11). Der Web-Server 2 lädt oder erzeugt diese Web-Seite (Schritt S12), die dann an den Client 1 via HTTP zurückgeschickt wird (Schritt S13) und vom Web-Browser des Clients 1 angezeigt wird (Schritt S14). Der Benutzer arbeitet mit der Web-Seite (Schritt S15), wobei in neuen Browsern wie IE 4/5, NS4/Mozilla Änderungen an den Web-Seiten via DHTML am Client durchgeführt werden können. Hierzu wird JavaScript/JScript und/oder JAVA verwendet. Die so am Client 1 geänderten Seiten können über diese Sprachen abgefragt werden. Bei IE4/5/Mozilla können Änderungen direkt über den Browser abgefragt werden, bei NS4 müssen Änderungen an der Web-Seite von einem entsprechenden Framework aufgezeichnet werden. Diese Aufzeichnung vom am Client 1 erstellten Änderungen erfordert die Verwendung einer umfangreichen Software, die zu Beginn einer Sitzung als JavaScript/VBScript und/oder Java-Applets vom Web-Server 2 geladen wird. Als Ergebnis wird jede Änderung am Client 1 aufgezeichnet. Die aufgezeichneten Änderungen werden in einem browserunabhängigen Format zum Web-Server 2 zurückgeschickt (Schritt S16), wobei das Zurückschicken der aufgezeichneten Änderungen über ein HTML-Formular, eine Java-Applet oder eine ActiveX-Control oder andere mögliche Mechanismen der Kommunikation zwischen Client 1 und Web-Server 2 erfolgt. Die an den Web-Server 2 zurückgeschickten Änderungen werden dort unter einer Client-Identität wie z.B. Sitzungsidentität (engl.: SessionId(entity)) bestehend aus Benutzeridentität (engl.: UserId(entity)) und Passwort in einer Datenbank 3 abgelegt (Schritt S17). Hat die Identifikation des Clients 1 noch nicht stattgefunden, kann an dieser Stelle eine Anmel-

dung über UserId und Passwort oder andere Mechanismen erfolgen. Der Web-Server 2 erzeugt für diesen Client 1 einen eigenen Bereich in der Datenbank 3, in der die benutzerspezifischen Daten abgelegt werden. Dieser Bereich wird als "Push-Back-Bereich" bezeichnet. Als Übertragungsprotokoll kann HTTP verwendet werden, wobei der Port des Web-Servers 2 oder ein eigener Datenport verwendet werden kann.

**Fig. 2** beschreibt das Szenario, welches zwischen dem Client 1 und dem Web-Server 2 beim Zweitkontakt bzw. bei jedem weiteren Kontakt zeitlich abläuft.

Der Client 1 stellt eine HTTP-Anfrage nach einer Web-Seite an den Web-Server 2 und identifiziert sich dabei am Web-Server 2 z.B. über UserId und Passwort (Schritt S21). Die Identifikation kann über UserId und Passwort, Cookies oder andere Techniken wie Smartcards erfolgen. Der Web-Server 2 prüft, ob für diesen Client bzw. Benutzer in der Datenbank 3 bereits ein Push-Back-Bereich eingerichtet ist. Wenn nein, handelt es sich um einen Erstkontakt, und die oben beschriebenen Schritte S12 bis S17 werden ausgeführt. Wenn ja, prüft der Web-Server 2, ob die Anfrage aus dem Push-Back-Bereich der Datenbank 3 geladen werden kann. Wenn nein, wird wie beim Erstkontakt eine generische Web-Seite an den Client 1 zurückgeschickt. Ist ein Push-Back-Bereich vorhanden, werden die client-spezifischen Daten geladen (Schritt S22) und im generischen XML/JavaScript-Format an den Client 1 versendet (Schritt S23). Alternativ kann auch reines HTML im Push-Back-Bereich gespeichert und an den Client 1 zurückgesendet werden, allerdings ist dann im allgemeinen kein Datenaustausch zwischen verschiedenen Browser-Versionen mehr möglich. Am Client 1 werden diese Daten über JavaScript oder Java interpretiert und dynamisch HTML-Code am Client 1 erzeugt (Schritt S24), wobei hier Browser-Spezifika abgehan-

delt werden. Über den Browser kann der Benutzer mit der Web-Seite (Schritt S25) arbeiten. Die vom Benutzer gemachten Änderungen werden wie beim Erstkontakt zum Web-Server 2 zurückgeschickt (Schritt S26) und dort unter der Client-Identität in der Datenbank 3 gespeichert (Schritt S27).

EPO - Munich  
20

29. Juli 2000

Stuttgart, 28.07.2000  
P7552EP Rk/pa

Patentansprüche

1. Kommunikationsverfahren zwischen einem Netzwerk-Client (1) und einem Netzwerk-Server (2), wobei ein vom Netzwerk-Client (1) angefragtes Netzwerk-Dokument vom Netzwerk-Server (2) an den Netzwerk-Client (1) geschickt und dann von einem Browser des Netzwerk-Clients (1) dem Benutzer zur Bearbeitung angezeigt wird, dadurch gekennzeichnet,  
daß die vom Benutzer am Netzwerk-Dokument am Netzwerk-Client durchgeführten Änderungen und Ergänzungen am Netzwerk-Client (1) über Software aufgezeichnet und im Netzwerk-Server (2) als benutzerspezifische Daten gespeichert werden und daß beim erneuten Aufruf des Netzwerk-Dokuments durch den Benutzer das bei einem früheren Aufruf am Netzwerk-Client (1) durch die Interaktion des Benutzers erstellte Netzwerk-Dokument anhand den gespeicherten benutzerspezifischen Daten wiederhergestellt wird.
  
2. Kommunikationsverfahren nach Anspruch 1, dadurch gekennzeichnet, daß die vom Benutzer am Netzwerk-Dokument durchgeführten Änderungen und Ergänzungen zunächst am Netzwerk-Client (1), insbesondere im Hauptspeicher oder auf der Festplatte, zwischengespeichert werden.
  
3. Kommunikationsverfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß der Netzwerk-Server (2) für die benutzerspezifischen Daten einen eigenen Bereich in einer Datenbank (3) anlegt.

4. Kommunikationsverfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß beim erstmaligen Aufruf eines Netzwerk-Dokuments eine Identifikation des Netzwerk-Clients (1) bzw. des Benutzers erfolgt.
5. Kommunikationsverfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das vom Benutzer geänderte bzw. ergänzte Dokument am Netzwerk-Client (1) aufgezeichnet und im Netzwerk-Server (2) in einem browserunabhängigen Format, insbesondere in XML, gespeichert wird.
6. Kommunikationsverfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß am Netzwerk-Client (1) die gespeicherten benutzerspezifischen Daten des Netzwerk-Servers (2) interpretiert und daraus ein für den Netzwerk-Browser lesbare Format erzeugt wird.
7. Computerprodukt, das geeignet ist, alle Schritte des Kommunikationsverfahren nach einem der Ansprüche 1 bis 6 durchzuführen, wenn das Programm auf einem Computer läuft.
8. Computerprodukt nach Anspruch 7 dadurch gekennzeichnet, daß das Computerprodukt ein Computerprogramm ist.
9. Computerprodukt nach Anspruch 7 dadurch gekennzeichnet, daß das Computerprodukt ein computerlesbares Speichermedium ist.

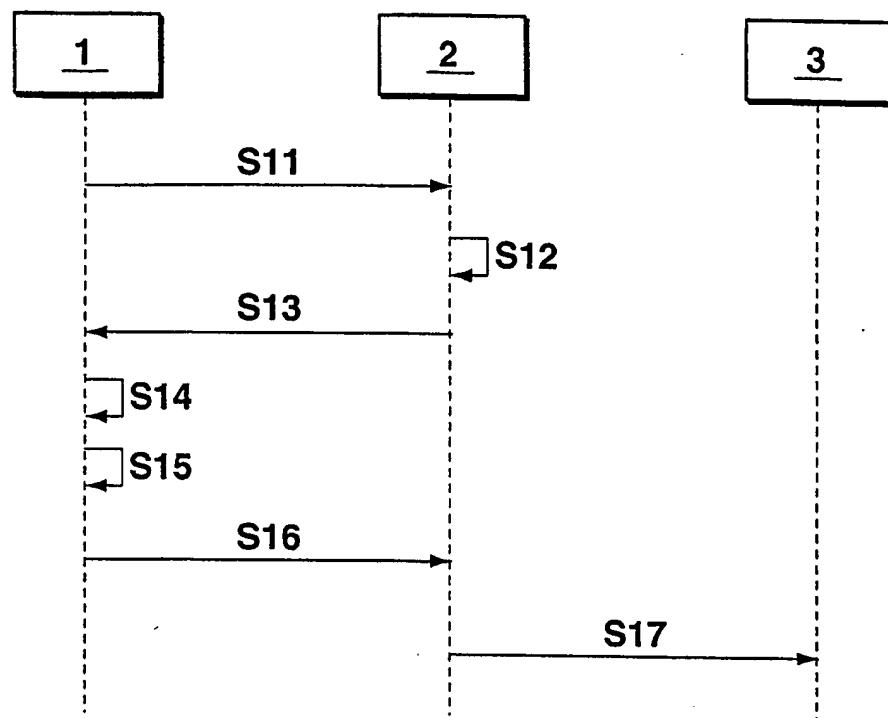
EPO - Munich  
20  
29. Juli 2000

Zusammenfassung

Bei einem Kommunikationsverfahren zwischen einem Netzwerk-Client und einem Netzwerk-Server, wobei ein vom Netzwerk-Client angefragtes Netzwerk-Dokument vom Netzwerk-Server an den Netzwerk-Client geschickt und dann von einem Browser des Netzwerk-Clients dem Benutzer zur Bearbeitung angezeigt wird, werden die vom Benutzer am Netzwerk-Dokument am Netzwerk-Client durchgeführten Änderungen und Ergänzungen am Netzwerk-Client über Software aufgezeichnet und im Netzwerk-Server als benutzerspezifische Daten gespeichert. Beim erneuten Aufruf des Netzwerk-Dokuments durch den Benutzer wird das bei einem früheren Aufruf am Netzwerk-Client durch die Interaktion des Benutzers erstellte Netzwerk-Dokument anhand den gespeicherten benutzerspezifischen Daten wiederhergestellt. So kann jedem Client beim erneuten Aufruf eines Netzwerk-Dokuments seine individuelle Netzwerk-Seite angeboten werden.

EPO - Munich  
20  
29. Juli 2000

**Fig. 1**



**Fig. 2**

